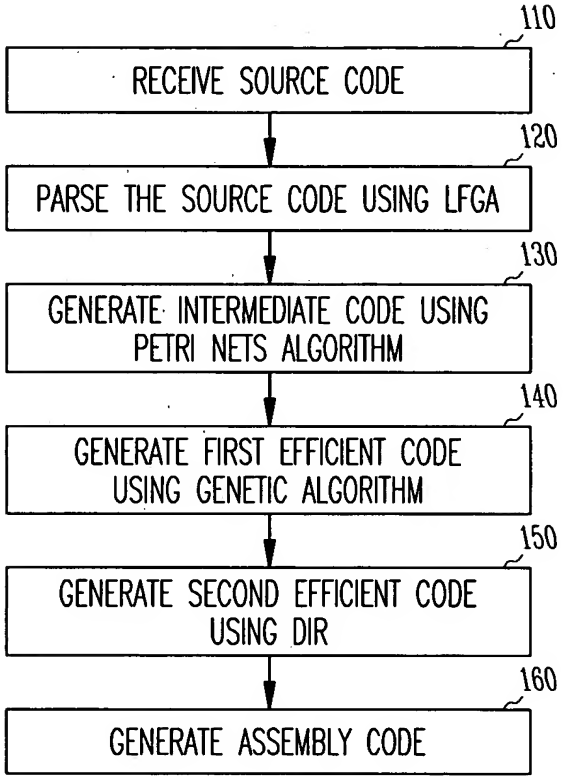
100
110
RECEIVE SOURCE CODE
120
PARSE THE SOURCE CODE USING LFGA
130
GENERATE INTERMEDIATE CODE USING PETRI NETS ALGORITHM
140
GENERATE FIRST EFFICIENT CODE USING GENETIC ALGORITHM
150
GENERATE SECOND EFFICIENT CODE USING DIR
160
GENERATE ASSEMBLY CODE

Fig. 1

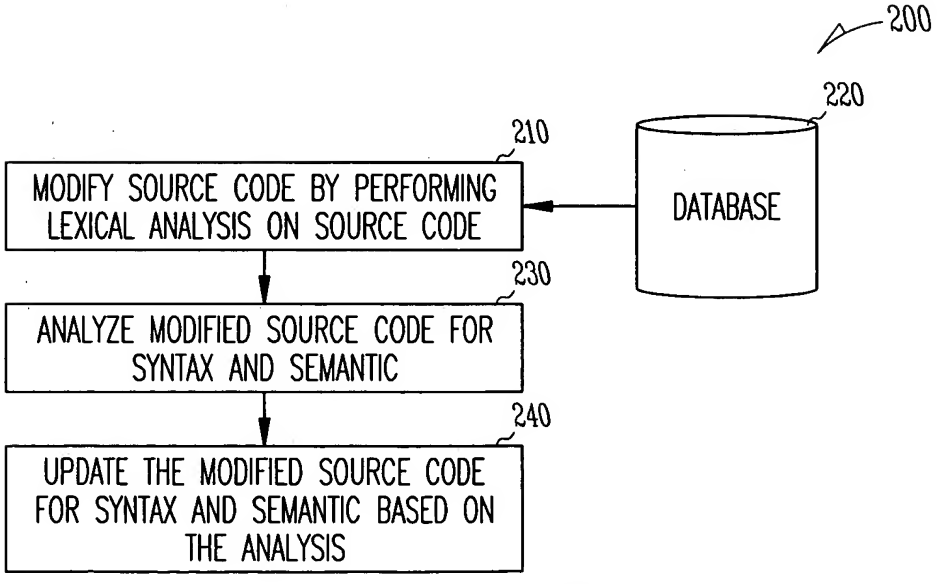
200
220
DATABASE
210
MODIFY SOURCE CODE BY PERFORMING LEXICAL ANALYSIS ON SOURCE CODE
230
ANALYZE MODIFIED SOURCE CODE FOR SYNTAX AND SEMANTIC
240
UPDATE THE MODIFIED SOURCE CODE FOR SYNTAX AND SEMANTIC BASED ON THE ANALYSIS

Fig. 2

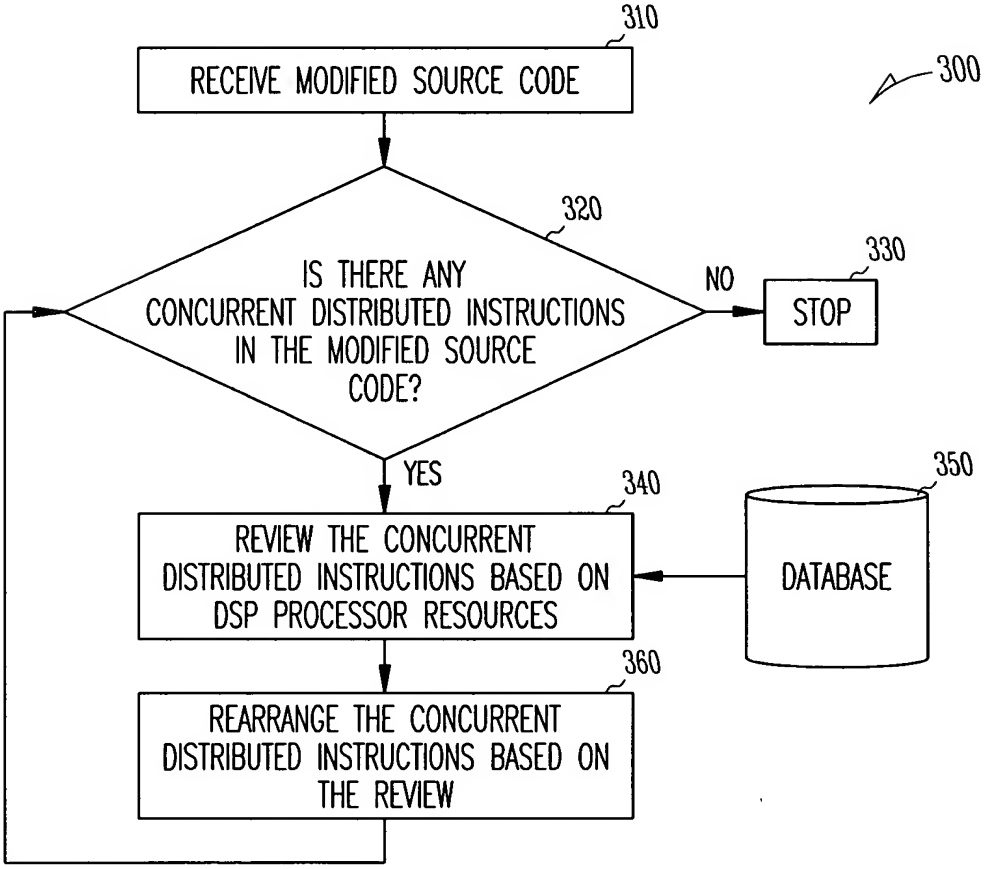
310
RECEIVE MODIFIED SOURCE CODE
300
320
IS THERE ANY CONCURRENT DISTRIBUTED INSTRUCTIONS IN THE MODIFIED SOURCE CODE?
NO
330
STOP
YES
340
REVIEW THE CONCURRENT DISTRIBUTED INSTRUCTIONS BASED ON DSP PROCESSOR RESOURCES
350
DATABASE
360
REARRANGE THE CONCURRENT DISTRIBUTED INSTRUCTIONS BASED ON THE REVIEW

Fig. 3

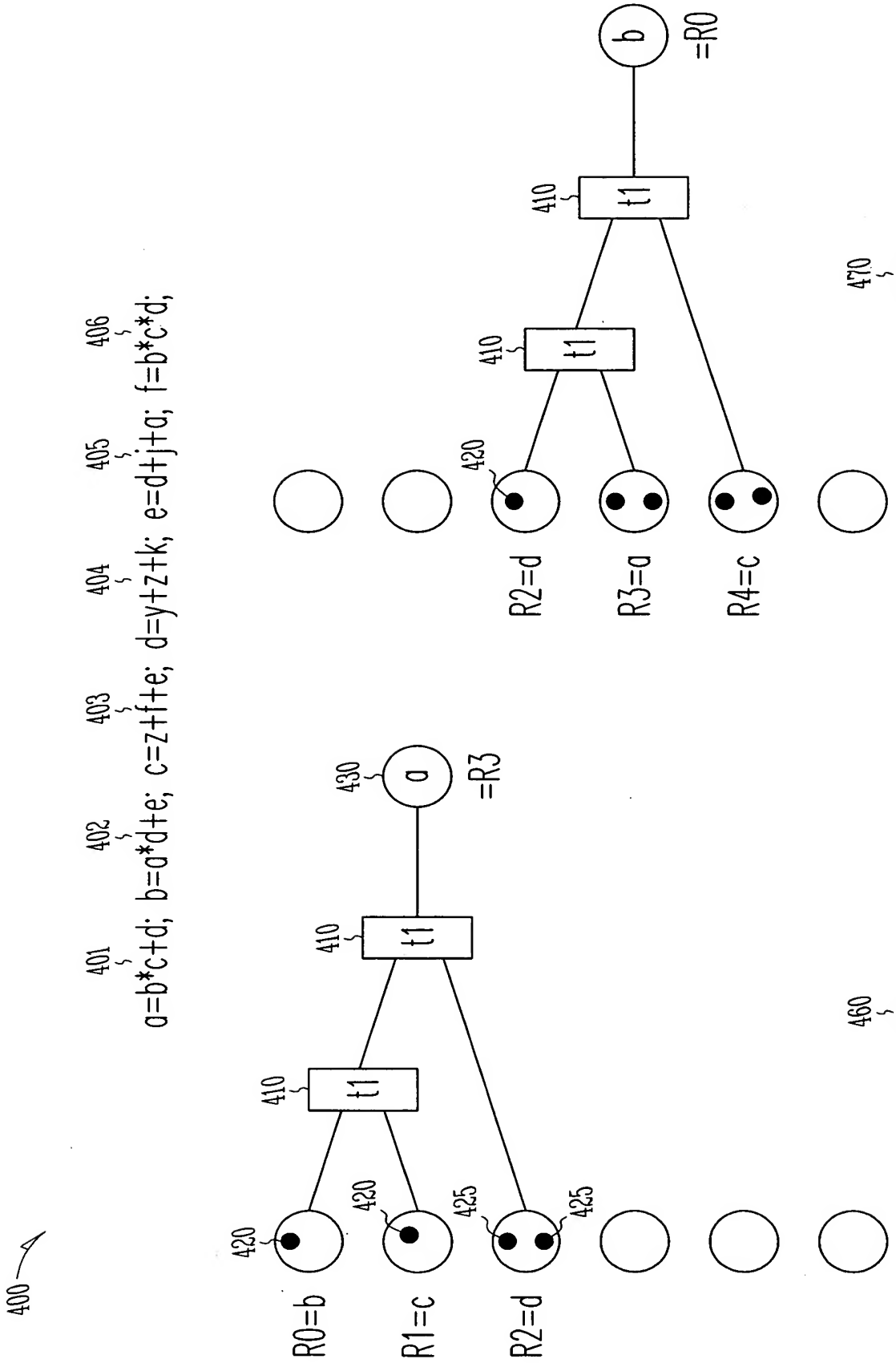
400
401
a=b*c+d; b=a*d+e; c=z+f+e; d=y+z+k; e=d+j+a; f=b*c*d;
402
403
404
405
406
460
STEP 1
470
STEP 2
410
t1
420
425
430
a
b
R0=b
R1=c
R2=d
R3=a
R4=c
=R3
=R0

Fig. 4

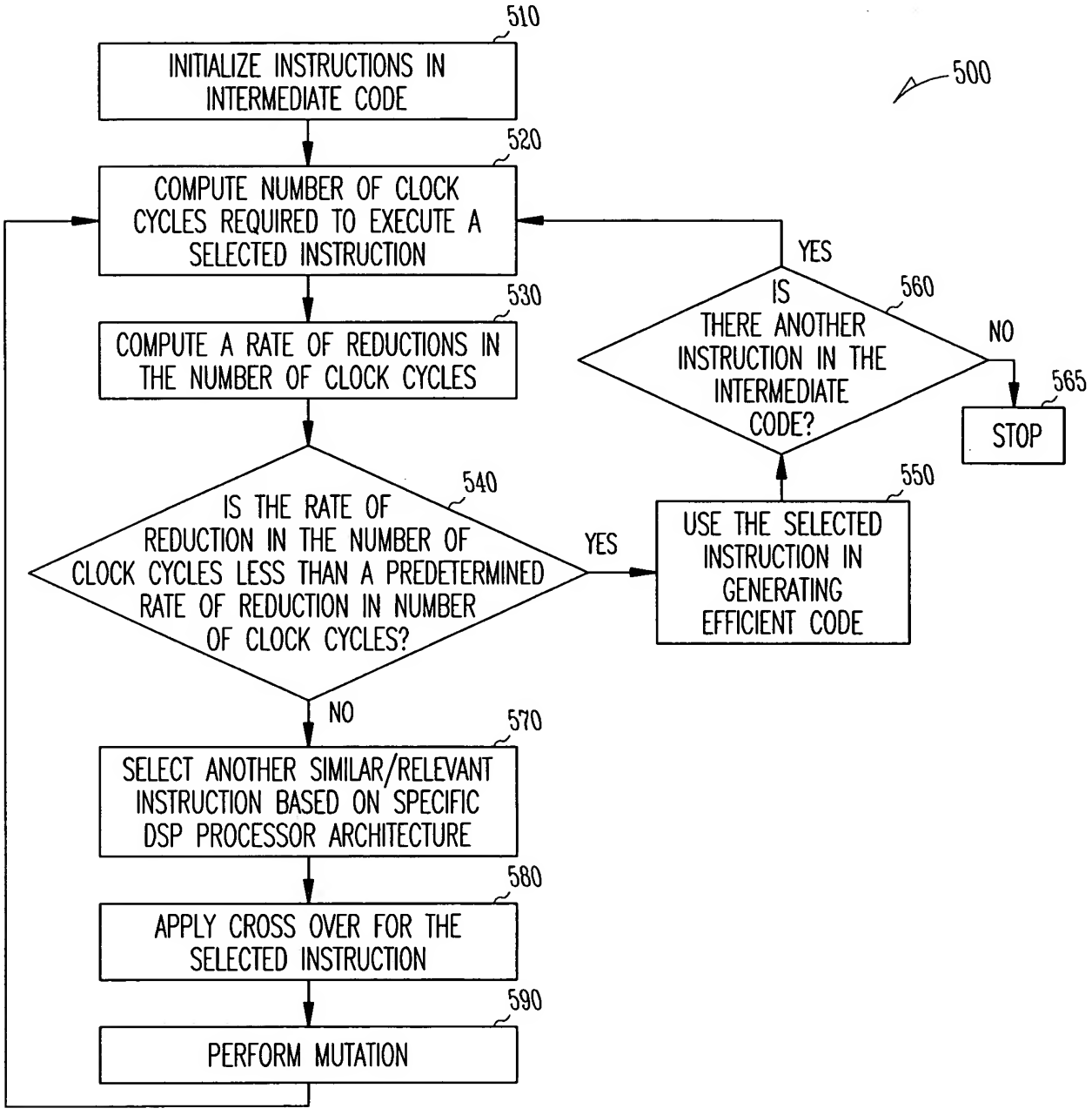
500
510
INITIALIZE INSTRUCTIONS IN INTERMEDIATE CODE
520
COMPUTE NUMBER OF CLOCK CYCLES REQUIRED TO EXECUTE A SELECTED INSTRUCTION
530
COMPUTE A RATE OF REDUCTIONS IN THE NUMBER OF CLOCK CYCLES
540
IS THE RATE OF REDUCTION IN THE NUMBER OF CLOCK CYCLES LESS THAN A PREDETERMINED RATE OF REDUCTION IN NUMBER OF CLOCK CYCLES?
YES
NO
550
USE THE SELECTED INSTRUCTION IN GENERATING EFFICIENT CODE
560
IS THERE ANOTHER INSTRUCTION IN THE INTERMEDIATE CODE?
YES
NO
565
STOP
570
SELECT ANOTHER SIMILAR/RELEVANT INSTRUCTION BASED ON SPECIFIC DSP PROCESSOR ARCHITECTURE
580
APPLY CROSS OVER FOR THE SELECTED INSTRUCTION
590
PERFORM MUTATION

Fig. 5

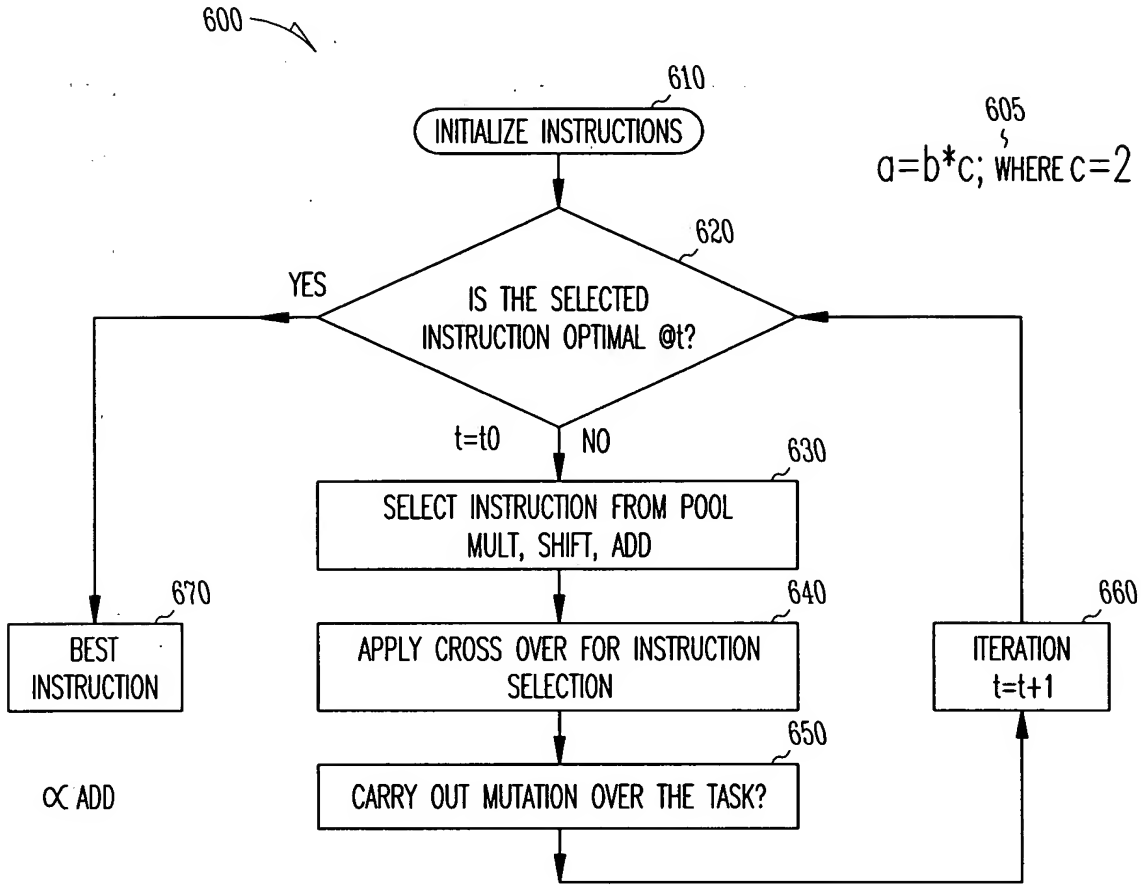
600
610
INITIALIZE INSTRUCTIONS
605
a=b*c; WHERE c=2
620
IS THE SELECTED INSTRUCTION OPTIMAL @t?
YES
t=t0
NO
630
SELECT INSTRUCTION FROM POOL MULT, SHIFT, ADD
670
BEST INSTRUCTION
640
APPLY CROSS OVER FOR INSTRUCTION SELECTION
660
ITERATION t=t+1
650
CARRY OUT MUTATION OVER THE TASK?
∝ ADD

Fig. 6

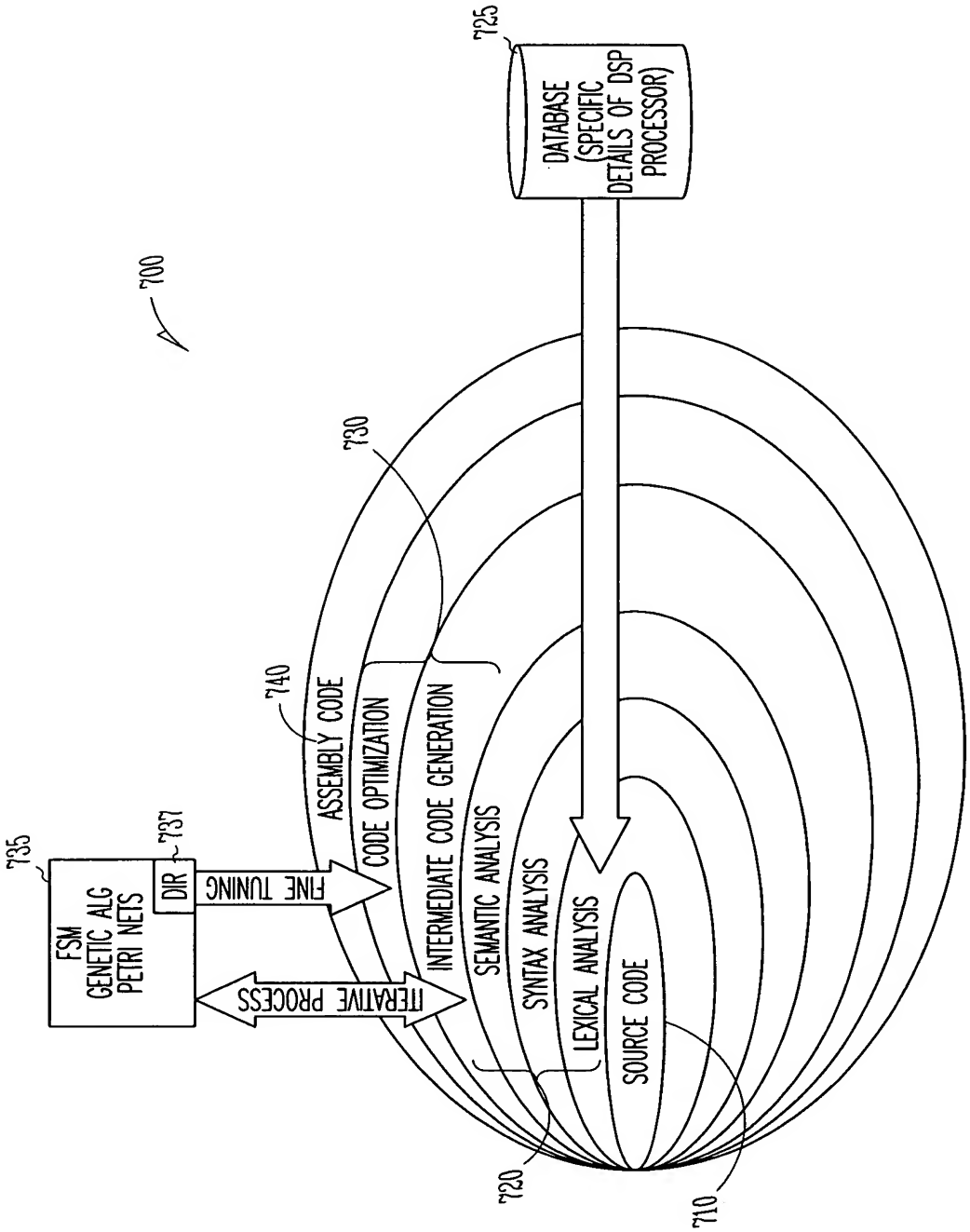
725
DATABASE (SPECIFIC DETAILS OF DSP PROCESSOR)
700
730
740
ASSEMBLY CODE
CODE OPTIMIZATION
INTERMEDIATE CODE GENERATION
SEMANTIC ANALYSIS
SYNTAX ANALYSIS
LEXICAL ANALYSIS
SOURCE CODE
735
FSM
GENETIC ALG
PETRI NETS
737
DIR
FINE TUNING
ITERATIVE PROCESS
720
710

Fig. 7

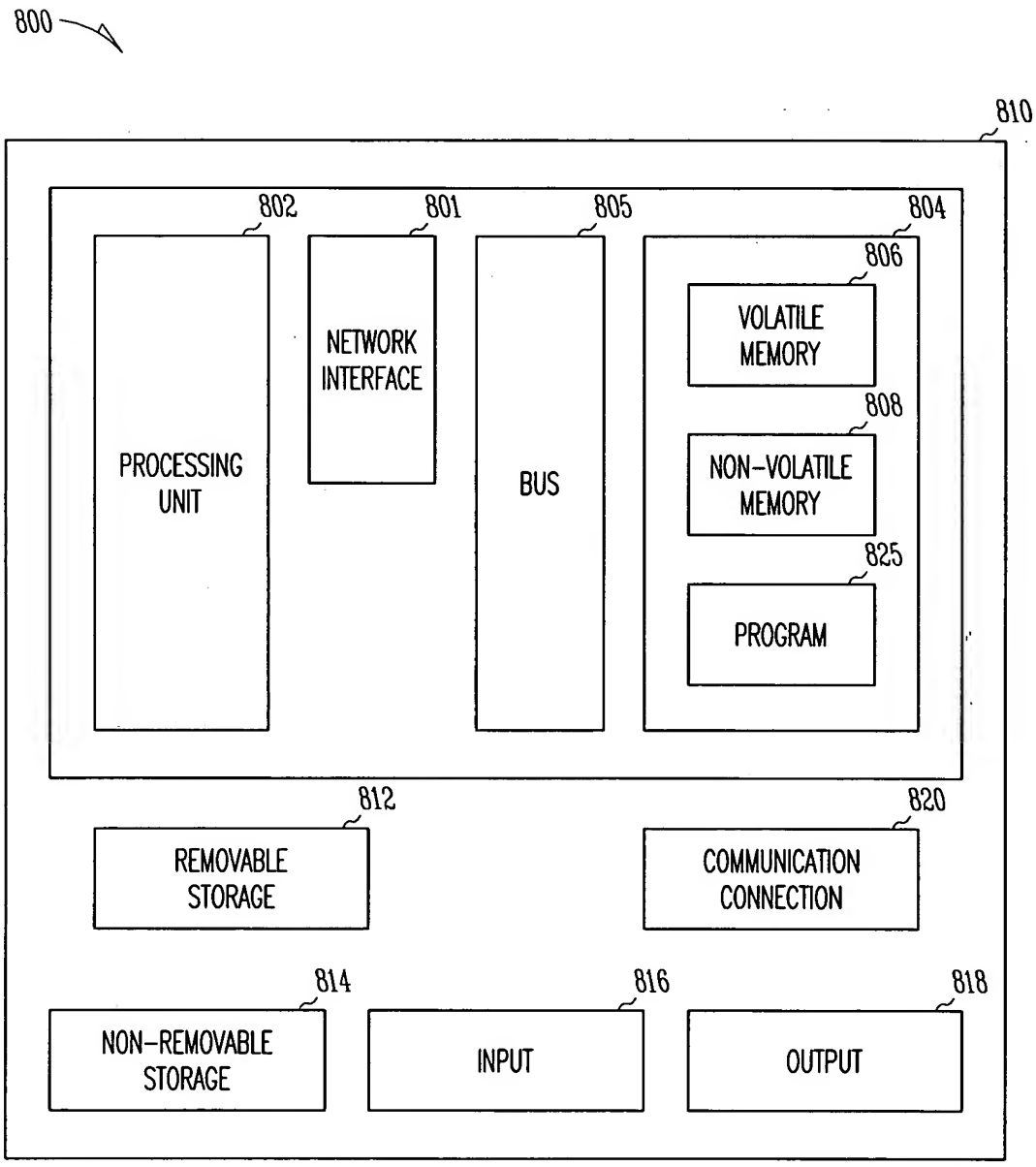
800
810
802
801
805
804
806
PROCESSING UNIT
NETWORK INTERFACE
BUS
VOLATILE MEMORY
808
NON-VOLATILE MEMORY
825
PROGRAM
812
820
REMOVABLE STORAGE
COMMUNICATION CONNECTION
814
816
818
NON-REMOVABLE STORAGE
INPUT
OUTPUT

Fig. 8